Electronic Version
Stylesheet Version v1.1.1

Description

AUTOMATIC LATCH COMPRESSION/REDUCTION

BACKGROUND OF INVENTION

[0001] Field of the Invention

[0002] The present invention generally relates to a method of designing an integrated circuit having latches and more particularly to a method which eliminates redundant latches by combining latches.

[0003] Description of the Related Art

[0004] Current designs consume a significant portion of their total power in the latches and clock tree. Designers require latches to hold the state of the design from cycle to cycle. While some clock gating has been added to the architecture, the sheer number of latches impact the area and the overall clock tree design. On the other hand, the actual use of particular latches on a cycle by cycle basis is often sparse. In addition, whole cores of latches may be non-

overlapping in a time domain. Unfortunately, designers are unable to effectively determine which latches are non-overlapping, especially, when often the overlapping latches are from diverse areas of the design and the designer may be unable to understand the uses of another designer's latch implementation.

Summary of Invention

[0005] The following disclosure presents a method of designing an integrated circuit having latches. The invention first prepares a logical design of logic devices and latches and then creates a physical design by positioning the logic devices and the latches within the integrated circuit based on the logical design. During the process of creating the physical design, the invention eliminates redundant latches by combining latches which do not transition during the same clock cycle, do not relate to the same logical function, are in the same clock domain, and are within a given physical proximity of each other.

[0006] The invention determines whether latches transition during the same clock cycle by running a simulation of an initial physical design and recording the latches that transition during each clock cycle. The invention also determines whether an adequate timing slack exists between

transitions of latches that do not transition during the same clock cycle.

[0007] The foregoing process of eliminating redundant latches comprises replacing at least two latches with a single latch. The process of eliminating redundant latches produces a revised physical design, and the invention tests the revised physical design to determine whether the revised physical design performs as expected. The invention determines whether the latches relate to the same logical function by recording which latches are associated with each logical function in the logical design and determines whether the latches are in the same clock domain by recording which latches are associated with each clock domain in the logical design.

Brief Description of Drawings

[0008] The foregoing invention will be better understood from the following detailed description of preferred embodiments with reference to the drawings, in which:

[0009] Figure 1A is a schematic diagram of a initial physical design before elimination of redundant latches;

[0010] Figure 1B is a schematic diagram of a initial physical design after elimination of redundant latches; and

[0011] Figure 2 is a flow diagram illustrating a preferred method

of the invention.

Detailed Description

[0012] As mentioned above, designers are unable to effectively determine which latches are non-overlapping, especially when the overlapping latches are often from diverse areas of the design and the designer may be unable or not understand the uses of another designer's latch implementation. The invention provides an automated approach of determining which latches can be shared across a design. As such, this invention provides a new method of reducing the number of latches within the design without impacting the functionality of the design.

[0013] Though simulation, the invention identifies the latches that are mutually exclusive functionally and combines them at the PD (physical design) backend. The exact latches to be combined is not known until PD and is not determined at architecture definition or even front end processing. Therefore, the invention is very suitable for control logic.

[0014] The latches are created manually during logic design by the designer for each function. Often teams of logic designers work together on the macros and tens of thousands of latches are being coded by a multitude of de-

signers. Each design team is focused on their scope of work and cannot be fully aware of each other's use of latches. In addition, a single designer can be responsible for 50 to 100 thousand lines of code, with often three to four thousand latches. Due to the nature of the design, the designer works on one portion of the design at a time and does not have time to fully determine which latches can be shared with others within his scope of design. Determining latches that can be shared outside the scope of a designer is not truly feasible. In addition, the latches that can be shared are for boolean non-related function and a human is unable to detect or realize whether or not latches have an overlapping time.

[0015] Though the use of simulation scoreboarding, the invention builds a database that determines (over the full simulation suite) the register usage over time. Essentially, for each time increment, a list of latches that require transition is maintained in the database. Once the simulation completes, a post processing program is run that updates a master list of registers and keeps a list per register of possible candidates. To simplify the process, in one embodiment, only latches within the same clock domain may be combined. However, the invention is equally applicable

to multiple clock domains.

[0016] Once the design is frozen and entering the physical design phase, the grouping of latches is based upon two additional criteria. First is the floor plan proximity of the function and the second is the checking of sufficient slack to support the automatic combination of functional latches. After identifying which latches are to be combined, an ECO (engineering change option) is applied to the gate level netlist to perform this compression.

[0017] The recombination is not limited to just two latches, the invention creates a linked list time domain structure such that more than two latches can be combined into a single latch, as long as the sequential execution of each function is non-overlapping.

[0018] Once the design has been modified, several techniques exists to verify that the altered design matches the original design. On the first order, the same architecture test-cases can be rerun to ensure that the results are the same. Some modifications may be needed if the verification calls out explicit registers that have been merged, then the new structure elements (the equilivance stage), must be referenced instead. Another test method is to use some of the standard formal verification tools. These tools

will verify a boolean equivalence and may be the best tool to ensure that the designs are exactly equivalent.

[0019] Figures 1A-1B illustrate the operation of the invention schematically. More specifically, in Figure 1A registers 10-13 utilize various combinational logic units 14, 15, which in turn utilize latches 16, 17. Note that register B utilizes both sets of combinational logic 14, 15. However, because register B can only utilize one of the sets of combinational logic 14, 15 at a time, latches 16 and 17 will never be used simultaneously. Therefore, assuming that there are no issues relating to logical function, clock domain, physical proximity, and assuming that there is adequate timing slack, latches 14 and 15 can be replaced with a single latch 20, as shown in Figure 1B. In the compressed example shown in Figure 1B, the combinational logic units 14, 15 act as a single unit and is therefore shown as a single unit 22. In addition, register B 11 is moved to a position downstream from the compressed latch 20 such that when register B would have previously utilized combinational logic 14, latch 20 acts like previous latch 16; and when register B would have previously utilize combinational logic 15, latch 20 acts like previous latch 17. Therefore, the replacement of the multiple

latches 16, 17 with a single latch 20 is transparent to those devices outside this circuit because the circuit still maintains two separate outputs that would output the same information in an identical manner.

[0020] The selection logic 23 locks non-active output(s) to a known logical state when a corresponding output (e.g., the other output(s) in a pair (or set) of outputs) is active. Essentially, the inverter 24 makes the output from the two AND devices 25 mutually exclusive. Similar arrangements with other logical devices to achieve the same result could be substituted here as would be understood by one ordinarily skilled in the art. The selection logic has more outputs than latches and preferably includes outputs equal in number to the number of latches in the initial physical design to allow the delay device 21 and selection logic 23 to provide a similar output as would have been seen with the non-compressed examples shown in Figure 1A. While only two outputs are illustrated in Figure 1B, as would be understood by one ordinarily skilled in the art in light of this disclosure, the selection logic 23 could (and would most likely) provide more than two outputs. Further, the inverter 24 combined with AND devices 25 (within the selection logic 23) permit one control signal to control a se-

ries of outputs similarly. Therefore, a single control signal could be used to control multiple sets of compressed latches within a hierarchical design.

[0021] Figure 2 shows the processing of the invention in flowchart form. In item 200, the invention pre-processes the scoreboard within the database by recording the name and clock phase of each latch. In item 202, the invention performs the above-mentioned simulation (by loading a first test case). Item 204 represents the simulation of one clock cycle and in item 206, the invention updates the scoreboard in the database to record which latches changed state in that cycle. The decision block 208 allows the invention to loop back through items 204 and 206 until the test case is complete. Similarly, item 210 loops back through the next test case until all test cases have been completed.

[0022] Once all test cases are completed, the invention performs a "post-process" on the scoreboard in the database which identifies latches that are candidates for compression based on clock domain and logical behavior. In item 216, the invention updates the scoreboard regarding the timing slack for each latch. Then, in the physical development stage shown in item, 218 the invention performs the se-

lection of the final compression groups based on slack, physical separation, etc.

[0023] Therefore, as shown above, the invention provides a method of designing an integrated circuit having latches. The invention first prepares a logical design of logic devices and latches and then creates a physical design by positioning the logic devices and the latches within the integrated circuit based on the logical design. During the process of creating the physical design the invention eliminates redundant latches by combining latches which do not transition during the same clock cycle, do not relate to the same logical function, are in the same clock domain, and are within a given physical proximity of each other.

[0024] The invention determines whether latches transition during the same clock cycle by running a simulation of an initial physical design and recording the latches that transition during each clock cycle. The invention also determines whether an adequate timing slack exists between transitions of latches that do not transition during the same clock cycle. The foregoing process of eliminating redundant latches comprises replacing at least two latches with a single latch. The process of eliminating redundant

latches produces a revised physical design, and the invention tests the revised physical design to determine whether the revised physical design performs as expected (e.g., performs as the non-compressed design would have performed).

[0025] While the invention has been described in terms of preferred embodiments, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.